Document Logical Structure Analysis Based on Perceptive Cycles

Yves Rangoni and Abdel Belaïd

Loria Research Center - Read Group, Vandœuvre-lès-Nancy, France
{rangoni,abelaid}@loria.fr
WWW home page: http://www.loria.fr/∼rangoni/
http://www.loria.fr/∼abelaid/

Abstract. This paper describes a neural network (NN) approach for document logical structure extraction. In this NN architecture, the document structure is stretched along the layers, allowing an interpretation decomposition from physical (NN input) to logical (NN output) level. The intermediate layers represent succesive interpretation steps. Each neuron is apparent (not hidden as in classical architectures) and is associated to a logical element. The recognition proceeds by repetitive perceptive cycles, propagating the information through the layers. In case of low recognition rate output , an enhancement is proceeded by error backpropagation leading to correct or choose a more adapted input feature subset. Several feature subsets hence are created using a modified filter method. The first experiments leaded on scientific documents are very encouraging.

1 Introduction

This paper addresses the problem of document logical structure extraction based on physical feature observations within document images. Although this problem have known a lot of solutions, it still remains very challenging for some specific noisy and variable document classes.

The literature abounds of various methods. A survey of the most important approaches in document structure analysis can be found in [1] for example. Most of them are based on formal grammars. However, these methods have drawbacks because the rules are given by the user and could be not sufficient to handle complex and noisy documents. It is difficult to remove ambiguities and a lot of thresholds must be fixed to process the matching between the physical and the logical structure.

Consequently, a method more oriented towards learning seems to be a more adapted solution. Artificial neural network (ANN) approaches allow such a training (rules are learnt) and are more robust to noise. However, ANN like the classical Multi Layer Perceptron (MLP) is considered as a black box and does not explicit the relationships between the neurons. In the same time, domain-specific knowledge appears essential for document interpretation as mentioned in [2] and

it seems useful to keep a part of knowledge in a Document Image Analysis (DIA) system.

In order to deal with theses two aspects (knowledge and learning), we propose a new ANN approach that use a Transparent Neural Network (TNN) architecture. This method will take MLP advantages and can act, in the same time, on the reasoning by introducing knowledge. The recognition task is done progressively by propagation of the inputs (local vision) towards the outputs (global vision). Back-propagation movements, during recognition step, are used for a input correction process as the human perception acts. These successive "perceptive cycles" (vision-interpretation) bring a context return which is very helpful for the input improvement.

This paper will be organized as follows. The first section is dedicated to the TNN architecture design. The second section will detail an input feature clusterization method to speed up the perceptive cycles and reproducing the human perception. Finally, the last section will be related to experimental results and discussions about the different methods proposed in this article.

2 The TNN Architecture Description

The proposed TNN architecture is described in Fig. 1. The first layer is made up of physical features where each element corresponds to a neuron. The following layers represent the logical structure at three different levels, from fine to

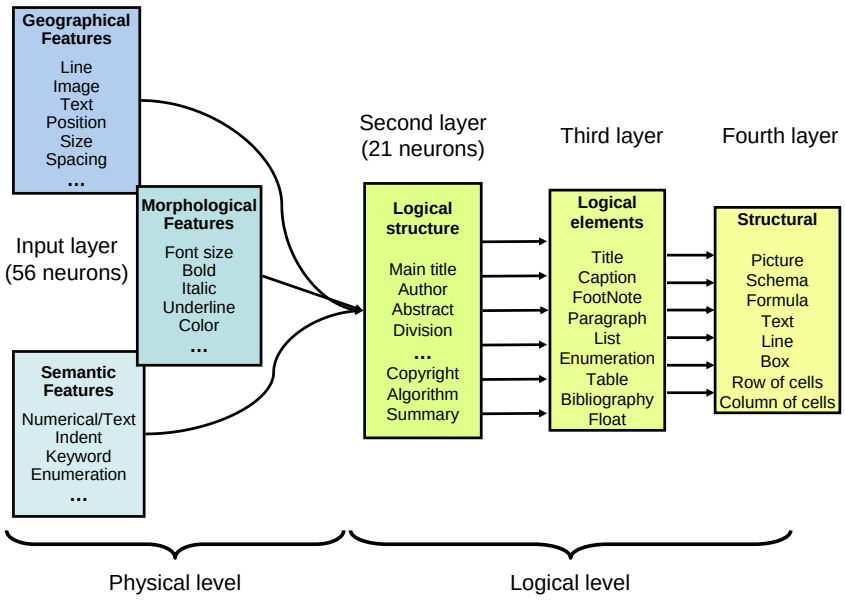


Fig. 1: Neuron semantic for document analysis.

coarse (see Fig.7 for the whole input and output names). All the layers are fully connected. This modeling integrates common knowledge on "general" document structure. It can be more precise if a DTD (Document Type Definition) is given as the DTD organizes the logical element in hierarchy. The real TNN ouput is the first logical level (the second layer) while the last layers represent the global context (third and fourth layers). There are used to precise the context that is needed for logical structure identification during the perceptive cycles.

This system can be considered as a hybrid method between a model-driven (i.e. DTD integration) and data-driven approach (i.e. training phase). As for a classical MLP, a database is used to train the links between physical and the final logical structure. The complete list of input and output is given in Fig. 7.

The training of the whole network is accomplished locally for each consecutive pair of layers. As the desired output of each layer is known, the system is seen as a MLP cascade without hidden layers. Weight modifications are carried out by an error correction principle. The first stage consists in initializing the weights with random values, then for each couple (Input, Output) of the training database, predicted value is computed by propagation. The error between the computed output P and the awaited output O is then determined. The second stage back-propagates this error in the previous layers. If the activation function is a sigmoid, the value to add to a weight $w_{i,j}$ is $\alpha(O_j - P_i)P_i(1 - P_j)I_i$.

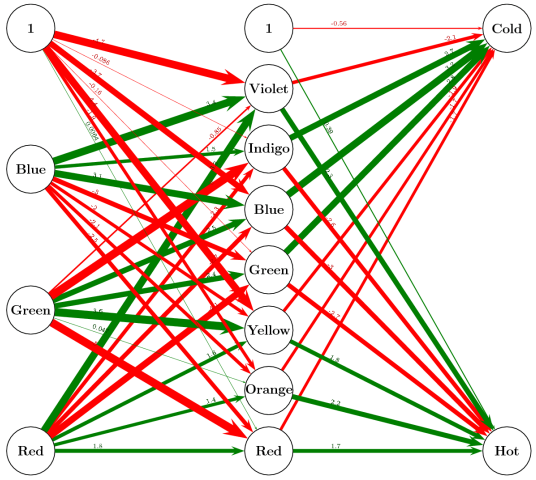


Fig. 2: A TNN classifying RGB pixels in hot and cold colors. It uses a rainbow color decomposition in its intermediary layer.

The Fig. 2 shows a small example of a trained TNN that classifies RGB colors in rainbow colors. The green links are for a positive contribution and red are for a negative one. The line thickness is proportional to the link weight.

Contrary to a MLP, the recognition process is more complicated. The MLP looks at the maximum output layer component O_i and deduces that the input pattern belongs the i^{th} class. In a TNN system, the outputs are analyzed and two decisions can be chosen (Fig. 3.):

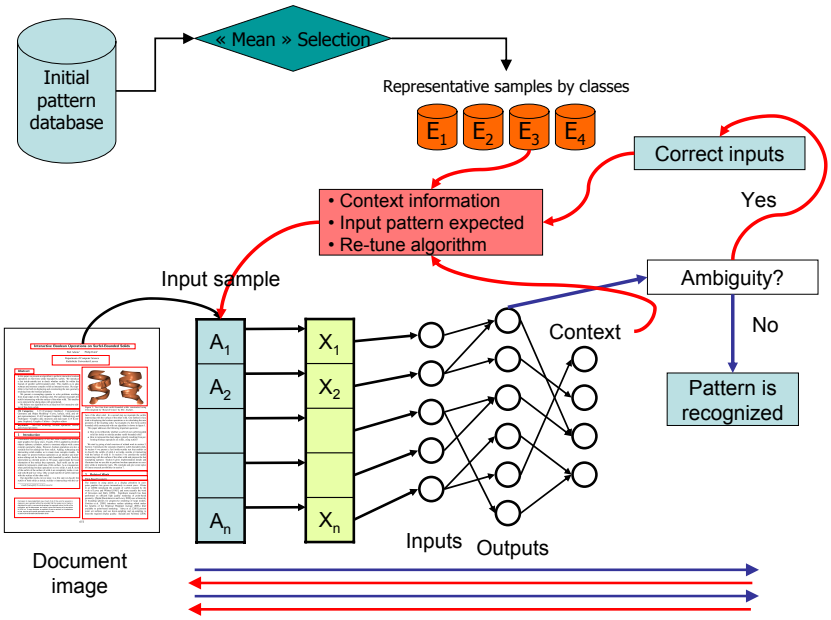


Fig. 3: Perceptive cycles: propagation, analyse, context return, correction.

- the first decision concerns the output when it is close to an unit vector. Thus, the system gives a ruling on a "good" pattern. This means that a class has a sufficient score $\|O\|_\infty \geqslant \varepsilon$ with $0 \gg \varepsilon > 1$ (acceptable class) and this winning class has a score greater than the others $\Gamma(O) = \frac{n((\sum O_i)^2 - \sum O_i^2)}{(n-1)(\sum O_i)^2} \leqslant \eta$ with $0 < \eta \ll 1$ (superior class). If such an output checks these rules, the system stops and the pattern is classified.
- the alternative decision occurs when the system reports an ambiguity (i.e. the pattern is confused among several classes). At that moment, the latest TNN layers react and propose a context. Thanks to the known neuron semantic, information from upper layers are used to determine the possible or unlikely classes. A hypothesis is created about the possible pattern class, then the input is analyzed in order to find the wrong component values.

As the input physical features (e.g. bounding box, font style, text, etc.) are determined by specific algorithms, it is possible to operate on their precision (or

quality) by reconsidering the algorithm parameters, or by changing totally the algorithm method. An example of "re-tuning" can be the OCR settings that give the text. It is possible in an OCR engine to change the amount of computation but change consequently the recognition quality. The "High Speed" mode is chosen when it is needed to separte text and image whereas "High Quality" mode is prefered if a precise word (a key-word for example) is searching in the text bloc.

Another example of algorithm "swapping" is the evaluation of word number in a text bloc. Two solutions can be chosen. The first algorithm uses a RLSA and evaluate the number of connex components. The second algorithm uses an OCR and simply counts the number of words. The first solution is the fastest but gives approwimates results whereas the second solution is more time expensive but is more accurate.

With the use of context, new information comming from the training database can be added during the correction. For example, if a segmentation problem occurs, the system find the "mean" awaited bounding box and corrects the previous bounding box dimensions. This example is not insignificant, because segementation error are frequent and penalize the whole physical extraction. The context returns allow often a better segmentation and contribute to a better recognition accuracy (see Fig. 4).

3 Input Feature Clustering

In the previous section, the TNN is showed to be able to analyze its outputs and improve its inputs accordingly. This system can obtain better scores than a MLP thanks to the perceptive cycles. However, "high-level" and time consuming features are needed to be extract for each cycle.

In order to speed-up the global process, the input features are categorized and classified in subsets. The feature subsets are used, progressively, as TNN input. A first feature set is chosen, then if the recognition rate is too low, another (containing the first and additional features) is selected and so on until reaching the final solution. As the whole features are not necessary needed to classify many patterns, the computation is consequently reduced.

Two criteria are used for feature classification: "quality" and "velocity". The "velocity" corresponds to the algorithm execution time given either by experiments or formally by studying its complexity. For the "quality" there is no straightforward measurement method. A specific method based on feature subset selection is proposed. The objective is to determine the best feature combination to feed a pattern classification system. This method is used to create a feature partitioning.

The literature mentions two main feature selection methods: filter and wrapper [3]. The first one selects variables by ranking them with correlation coefficients (it is usually suboptimal for building a predictor, particularly if the variables are redundant). The second one assess subsets of variables according

to their usefulness to a given predictor (but the predictor is needed to construct the subsets).

As the subsets are needed to construct the TNN architecture, a filter method has been considered. The filter method is also adapted to exclude many redundant variables in the same subset and keeping the most relevant ones.

The Karhunen-Loeve transform is used as a first step for the filter selection method. In [4] we used an extension of the PCA in order to build subset of initial features and not rewrite the features in another base (as the PCA is originally designed).

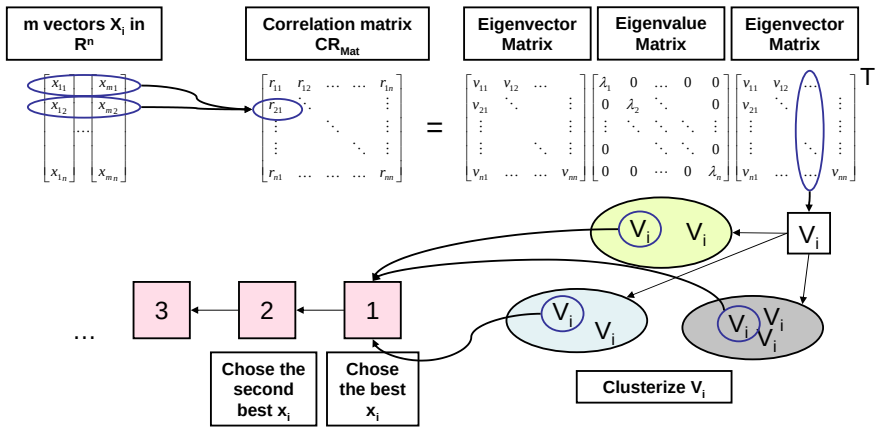


Fig. 4: Data categorization according to predictive capacity.

The eigenvectors V (in absolute value) of the data correlation matrix $CR_{Mat} = (cor(X_i, X_j))$ are computed. The vectors are then clusterized using a Self Organized Map (SOM) with an Euclidian distance (Fig. 4.). The clusters obtained contain similar eigenvectors (i.e. redundant variables). The feature corresponding to the nearest eigenvector from each cluster center is chosen to create a new subset. This subset contains high predictive features which are the less correlated. By fixing the SOM neuron number the number of desired subsets can be chosen.

An important phase of this clusterization process is to set the lower-space dimension q (i.e. the variance to be kept). As there is no optimal solution, some heuristics proposed by the literature have been tested:

- fixed number q: this is a straightforward method where cutting level is imposed by the user.
- fixed percentage: similarly to the previous case, but here the user choose the first $p\%$ of the eigenvalues.
- cumulated percentage: the number q is determined when the sum of the first variance (eigenvalue) is greater than a given fixed percentage.

These three methods are frequent but this assumes that the user overcome its application and can appreciate what dimension he must use. They are often used in social sciences because it is easier to interpret the data. Two other methods which are more general and more robust are founded on the shape of the eigenvalue sequence:

- Kaiser method: the average of all the variances is calculate, the space dimension q is determined when the sum of the first variance is greater than this average. Of a wide spread employment, it can be put at fault.
- Cattell method: [5] suggests to find the place where the smooth decrease of eigenvalues appears to level off to the right of the plot (the scree-test). This heuristic is often considered as the most powerful [6]

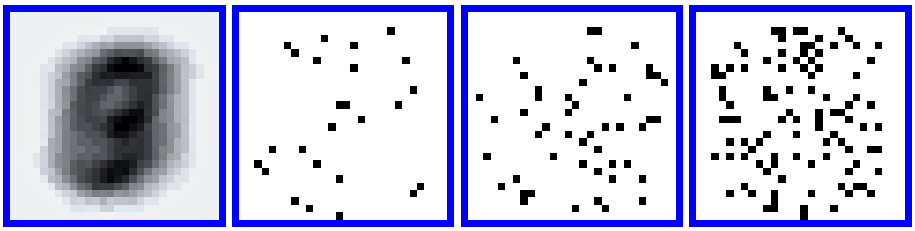
4 Experimental Results and Discussion

Before introducing results on document image anlaysis, experiments about "low-level" and highly correlated features are presented.

A first experimentation procedure is mainly employed to illustrate the variable subset creation method. We use the MNIST database [7].

A MultiLayer Perceptron is used to evaluate the group validity. This classifier has the same settings along all the experiments (topology, initial random weights, etc.). We have made two experiments have been made on this database The first uses the whole initial pixels of each image (digits are 28×28 pixel images). The second experiment uses resampled images (in a 7×7 format). Thus, we have for the first experiment 784 variables and for the second 49.

The MLP is trained with the different variable subsets and the recognition rate is chosen as a quality measurement. The subsets are compared to the randomly created. One thousand of random subsets have been generated and evaluated by the MLP. Then the best one is retained for the comparison. This procedure will be the same for the following experiments about document analysis.



Mean digit image | 25 chosen pixels | 50 chosen pixels | 100 chosen pixels

Fig. 5: Feature subsets created for MNIST database.

Table 1 shows normalized comparison results between the subset obtained by our method and the best of the random subsets for the initial MNIST database.

# features	Method	
	Random	Our selection
784(max)	100%	100%
500	98.4%	99.2%
300	95.9%	98.4%
150	90.5%	96.5%
100	84.2%	94.2%
50	70.9%	87.8%
25	47.1%	67.6%

Table 1: MNIST digit classification accuracy while decreasing the number of features

# features	Method	
	Random	Our selection
49(max)	100%	100%
35	94.2%	99.3%
25	81.2%	88.6%
15	56.2%	70.5%
10	43.9%	55.2%

Table 2: Resampled MNIST digit classification accuracy while decreasing the number of features

The Fig. 5 shows where the selected pixels are in the image. The first image represents the "mean" digit comming from the whole database ($\frac{1}{n}\sum_{i=1}^{n} I_i$) and in the next three pictures, chosen pixels can be seen. The table 2 is similar to the Table 1 but here, 7×7 pixel images are used for test.

The approach gives good results in spite of the strong influence of each pixel (expecting those on the border) on the classifier. The method keeps the two thirds of the information by keeping less than 4% of features (Table 1: with 25 of the 784 variables 67.6% of the information is kept).

Experiments concerning the document logical structure analysis are presented below. We have chosen as a main database some Siggraph 2003 conference papers [8]. The documents are scientific articles having numerous and diversified logical structure elements (see Fig. 6 for two examples).

In these 74 documents, 21 logical structures are labeled that represents more than 2000 patterns. The input and output features are presented in Fig. 7. Note that all the physical inputs (geometrical, morphological and semantic) are numerical values between 0 and 1 after possibly a normalization. In general, the number represents a percentage (e.g. the percentage of bold charaters in a

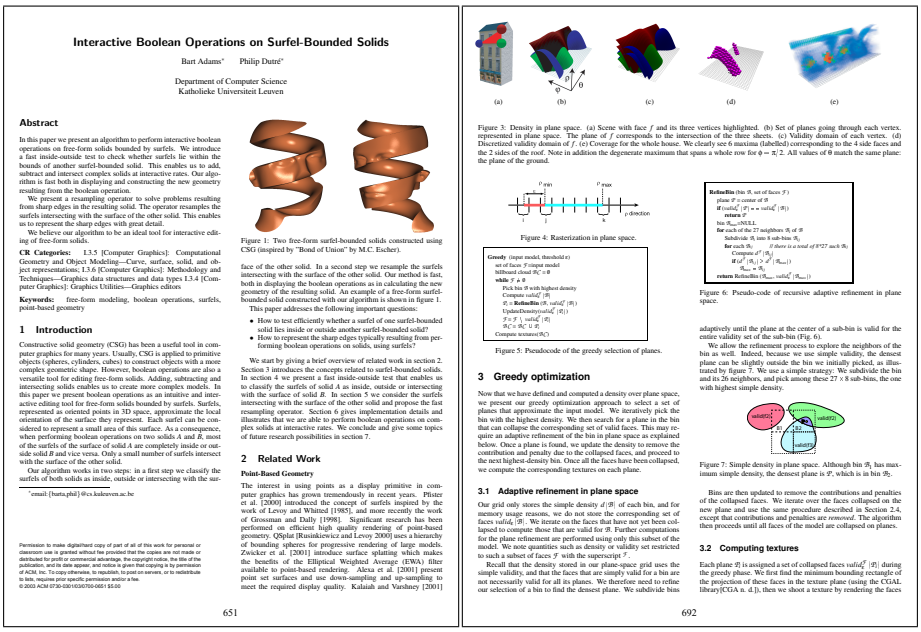


Fig. 6: Two scientific document database samples.

text bloc) and for other features that represent a number (e.g. the number k of keywords in a text bloc) we use the serie $\sum_{n=1}^{k} 1/(n+1)$ to have a number between 0 and 1.

As previously, the same input feature selection protocol is experimented on this document database. We extracted physical information from the document layout. There are 56 features holding geometrical, typographical, and morphological information (see Fig. 7) and we use once again a MLP as classifier.

Table 3 synthesizes some results of logical structures recognition accuracy according to the eigenvalue choice methods as mentioned at the end of the previous section. The five methods have been tested on different subset sizes.

The space dimension choice influences the results quality. Even if the MLP is a classifier able to give good results with few features, choosing too low or too high eigenvector dimension can be bad for the input feature clustering and consequently on the classifier.

It seems here (and for other tests that have be done on MNIST) that the Cattell method (that choose $q = 19$) is most of the time better than Kaiser (with $q = 14$). The two methods, which automatically find the number q, give the same or superior results than the classical ones where the operator must fix this number. We will hold for the following tests the Cattell method that seems to be the most robust on many experimentations.

Considering the results in Table 4, we observe that this "high-level" features better lends themself to this selection as we were expecting.

Logical	Physical		
	Geometrical	Morphological	Semantic
Title	Text	Bold	IsNumeric
Author	Image	Italique	KeyWords
Email	Table	Underlined	%KnownWords
Locality	Other	Strikethrough	%Punctuation
Abstract	x position	UpperCase	Bullet
Key words	y position	Small Capitals	Enum
CR Categories	Width	Subscript	Language
Introduction	Height	Superscript	Baseline
Paragraph	NumPage	Font	
Section	UpSpace	Font Size	
SubSection	BottomSpace	Scaling	
SubSubSection	LeftSpace	Spacing	
List	RightSpace	Alignment	
Enumeration		LeftIndent	
Float		RightIndent	
Conclusion		FirstIndent	
Bibliography		NumLines	
Algorithms		Boxed	
Copyright		Red/Green/Blue	
Acknowledgments			
Page number			

Fig. 7: Logical outputs and physical inputs for documents.

Choosing a small set of features here is more difficult. This method seems to be appropriate when the number of features is rather small and can be very powerful in this case (more than 83% of information is kept by dividing the variable number by 5).

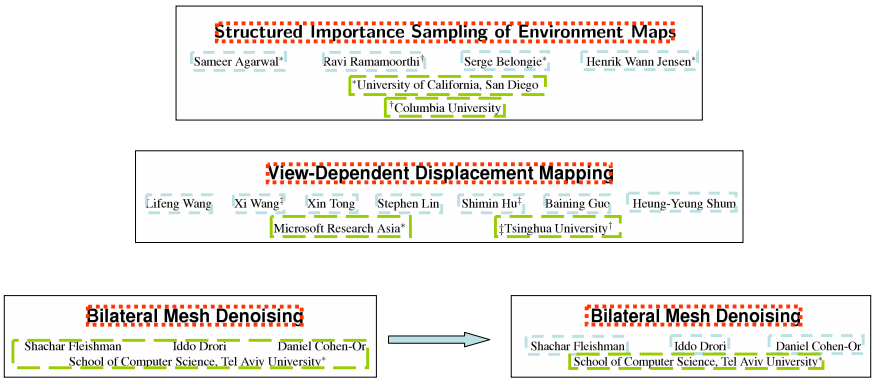
Feature number	Fixed number		Fixed %		% Variance		Kaiser	Cattell
	Num	Accur.	F%	Accur.	%V	Accur.	(q=14)	(q=19)
5	2	64.4	2%	61.6	10%	66.5	69.2	68.1
	5	64.3	5%	72.1	20%	67.9		
	10	60.3	10%	64.3	40%	61.4		
	15	59.2	20%	57.8	60%	63.6		
10	2	78.4		79.7		81.1	77.7	82.3
	5	79.8		82.7		73.5		
	10	72.9		77.1		78.4		
	15	70.0		76.6		72.6		
20	2	85.4		82.1		82.3	85.7	86.1
	5	84.9		82.8		86.1		
	10	83.6		83.3		82.3		
	15	82.6		83.3		78.8		
30	2	85.2		82.9		84.2	87.4	88.0
	5	86.8		85.6		85.8		
	10	86.6		86.5		86.7		
	15	86.3		85.4		87.7		

Table 3: Logical structure recognition accuracy (in %) according to dimensionnality q reducing method.

# features	Method	
	Random	Our selection
56(max)	100%	100%
35	86.9%	99.3%
25	65.0%	79.6%
15	51.8%	80.1%
10	35.1%	83.8%
5	17.9%	44.9%

Table 4: Logical elements classification accuracy while decreasing the number of features

Leaving side input features selection, results about complete DIA system are presented. Three inputs features subsets are created with the preceding method. Extraction tools, which can be configured, are used for extraction the physical layout. During the recognition phase, the sytem can chose between the feature subsets and act on extraction tools as mentioned in Section 3. The training stage uses 44 documents and 30 are used for the testing. Test results between a MLP and the TNN at the end of four perceptive cycles are presented in Figure 5.



The perceptive cycles increase the recognition rates (after 4 cycles the classifier reach 91.7%). A TNN without perceptive cycles is worse than a MLP (45.2% instead of 81.6%) because TNN does not have many constraints in its intermediate layers. With perceptive cycles, the context returns makes it possible to gain in precision while multiplying only, in our case, the computing time by about 2.5.

Recognition rates	MLP	TNN C_1	C_2	C_3	C_4
All elements	81.6%	45.2	78.9	90.2	91.7%
Best class	86.9%	66.7	85.3	85.3	99.3%
Worst class	0.0%	0.0	0.0	4.0	28.6%
Recognition time (MLP as reference)	100%	70%	145%	185%	240%

Table 5: Logical elements classification by MLP an TNN with perceptive cycles

5 Conclusions

We have presented in this article a neural network architecture for document logical structure analysis. The method uses a Transparent Neural Network that make possible to introduce knowledge in each neuron and organize in hierarchy the neurons in order to create a "vision" decomposition. The topology can simulate a decomposition hierarchy from fine (the patterns to recognized) to coarse (the global context). Thanks to this system, we can adapt the computation amount according to the pattern granularity and complexity. This "perceptive cycles" as named in cognitive psychology allows to simulate in the same system an recognition process that use automatic and fixed knowledge rules, a hierarchical view, and a interpretation-correction process thanks to hypothesis creation. An input feature clusterization has been made to speed-up the perceptive cycles.

The TNN gives encouraging results. Although some improvments are in hand, tests are already better than a simple MLP, without necessarily adding too heavy computations. In our future works, we will propose a genetic-method to choose representative samples in the database during the context return. Another works will be done to improve the feature subset creation and a method to deal with the final cases of rejected patterns will be presented.

References

1. Mao, S., Rosenfelda, A., Kanungo, T.: Document structure analysis algorithms: A literature survey. SPIE Electronic Imaging (2003)
2. Nagy, G.: Twenty years of document image analysis in pami. PAMI (2000)
3. Guyon, I., Elisseeff, A.: An introduction to variable and feature extraction. Journal of Machine Learning Research (2003)
4. Rangoni, Y., Belaïd, A.: Data categorization for a context return applied to logical document structure recognition. ICDAR (2005)
5. Cattell, R.: The scree test for the number of factors. Multivariate Behavioral Research (1966)
6. Zwick, W.R., Velicer, W.F.: Comparison of five rules for determining the number of components to retain. Psychological Bulletin (1986)
7. LeCun, Y.: (http://yann.lecun.com/exdb/mnist/)
8. Siggraph: http://www.siggraph.org/s2003/. (2003)